

# **Analýza a generování pravidel pro rozšíření PermissionACL systému MediaWiki**

## **Analysis And Creating Rules For PermissionACL MediaWiki Extension**

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Ostravě 10. dubna 2010

.....

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 10. dubna 2010

.....

Rád bych poděkoval vedoucímu bakalářské práce , Ing. Janu Vavříčkovi, za pomoc, věcné připomínky a veškerý čas, který mi věnoval.

## **Abstrakt**

Cílem této bakalářské práce je řešení problémů s vytvářením obecných pravidel pro řízení přístupů mediawiki serverů a vytvoření aplikace schopné interaktivně zobrazit a dále přetvářet obsah jakéhokoli textového vstupu obsahujícího daná pravidla. Výsledkem práce je nová aplikace pro vizualizaci a následné přetváření pravidel v co nejjednodušší a nejucelenější podobě pro potřeby široké uživatelské obce. Vizualizační aplikace je naprogramována v skriptovacím jazyce PHP.

**Klíčová slova:** Token, Framework, ACL, Wiki, PHP, HTML, CSS

## **Abstract**

The aim of this thesis is to resolve problems with the creation of general rules for access control mediawiki servers and create application that can interactively display and transform the content of any text entry containing the rule. The result of this work is a new application for visualization and subsequent transformation rules in the simplest and most complete form for the needs of a wide users community. Visualization application is programmed in PHP scripting language.

**Keywords:** Token, Framework, ACL, Wiki, PHP, HTML, CSS

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Programovací jazyk PHP</b>	<b>6</b>
2.1	Výhody . . . . .	6
2.2	Nevýhody . . . . .	6
<b>3</b>	<b>MediaWiki</b>	<b>7</b>
3.1	Seznámení s Wiki . . . . .	7
3.2	Práce s Wiki dokumentem . . . . .	7
3.3	Rozšíření PermissionACL . . . . .	9
<b>4</b>	<b>Praktická část</b>	<b>13</b>
4.1	Zadání . . . . .	13
4.2	Analýza . . . . .	13
4.3	Uložení editovaných pravidel . . . . .	19
4.4	Grafické zpracování . . . . .	20
4.5	Vývojové prostředí . . . . .	22
<b>5</b>	<b>Závěr</b>	<b>23</b>
<b>6</b>	<b>Reference</b>	<b>24</b>
	<b>Přílohy</b>	<b>24</b>
<b>A</b>	<b>Obsah přiloženého CD</b>	<b>25</b>

## Seznam tabulek

1	Výpis některých kódů parseru . . . . .	16
---	--	----

## Seznam obrázků

1	Mapa rozložení wiki serverů . . . . .	8
2	Vytvoření nové wiki stránky pokud nebyla nalezena ve vyhledávání . . . . .	8
3	Editace wiki dokumentu . . . . .	9
4	Ukázka využití ACL . . . . .	11
5	ACL Sekvenční diagram komunikace objektů . . . . .	12
6	Formulář pro vstupní data obsahující textová pravidla . . . . .	14
7	Obrázek s formulářem pro editaci pravidel . . . . .	17
8	Export pravidel v textové podobě . . . . .	17
9	Architektura JQuery . . . . .	19

## Seznam výpisů zdrojového kódu

1	Ukázka vstupních pravidel . . . . .	14
2	Ukázka části pole tokenů. . . . .	15
3	Průchod polem tokenů. . . . .	15
4	Metoda ověřující správnost tokenů. . . . .	15
5	Průchod polem tokenů. . . . .	16
6	Zpracování výpisu pravidel . . . . .	18
7	Ukázka připojení sortable na html element UL . . . . .	18
8	Ukázka serializace pravidel . . . . .	20
9	Ukázka css stylů . . . . .	21



## 1 Úvod

S rozvojem informačních technologií přichází doba, kdy se lidé snaží co nejvíce zjednodušit práci s počítači pro potřeby široké uživatelské obce. Jedná se zejména o zpříjemňování práce pomocí grafických uživatelských rozhraní. Práce se tak stává jednodušší a dostupnější i lidem, kterým stačí znát pouze základy zacházení s počítačem, aby si mohli např. spravovat vlastní účetnictví, číst emaily, nebo užívat si komunikaci s přáteli pomocí různých sociálních sítí.

Praktická část bakalářské práce se zabývá tvorbou aplikace, která zjednoduší a zpříjemní práci s vytvářením a editací pravidel pro rozšíření mediawiki serverů. Pro tvorbu jsem použil skriptovací jazyk PHP za použití javascriptového frameworku pro vytvoření uživatelsky příjemného rozhraní pro práci s daty.

## 2 Programovací jazyk PHP

Zkratka pro "Personal Home Page". PHP je skriptovací programovací jazyk využívaný především pro programování internetových stránek. PHP se často integruje do HTML kombinací těchto dvou technologií.

PHP umožňuje vývoj různorodého druhu aplikací počínaje jednoduchým počítačem přístupů a konče tvorbou podnikového informačního systému v prostředí Intranetu či Extranetu. Možnosti PHP ještě umocňuje důsledná integrace s mnoha databázovými systémy. Bez systému jako PHP by dnes nemohly pracovat virtuální obchodní domy, obchodní rejstřík na Internetu, rezervační systémy, nebo třeba prohledávání katalogu vašeho oblíbeného dodavatele dámského spodního prádla, počítačových komponent či hudebních nahrávek.[3]

### 2.1 Výhody

- Jednoduchost na naučení
- Velká podpora webhostingů
- Svobodná licence
- Rozsáhlá programátorská komunita
- Výborná dokumentace
- Multiplatformovost

### 2.2 Nevýhody

- Chybí podpora debugování v základu
- Špatná podpora integrace s jinými technologiemi
- V základu chybí nástroj pro dokumentaci

## 3 MediaWiki

MediaWiki je profesionálním wiki systémem původně vytvořeným pro potřeby webové encyklopedie Wikipedia. Je navržena pro práci s velkým množstvím uživatelů a stránek bez omezení výkonu a funkčnosti. Oblíbenost této wiki spočívá právě v její nepřekonané výkonnosti stejně jako v propracované struktuře stránek. [1]

K dosažení takového výkonu je zapotřebí odpovídající výpočetní síly, kterou zajišťují servery rozmístěné po celém světě. Hlavním sídlem je americká Florida, kde se nachází centrum webových serverů viz. obrázek č.1

Jednou z nejznámějších wiki, která je spuštěna na wikimedia je všem tolik známá encyklopedie Wikipedia.

### 3.1 Seznámení s Wiki

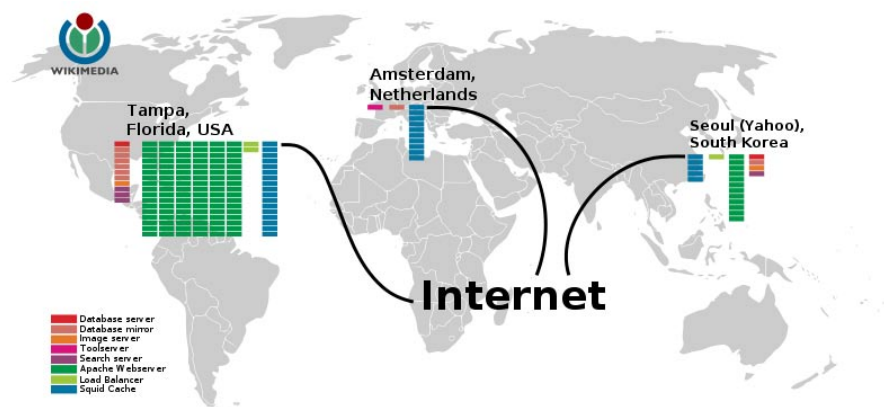
Wiki název pocházející z havajského slova "wiki" znamenající "rychlý" je systém umožňující uživatelům správu hypertextových dokumentů, pomocí kterého mohou přidávat nové, nebo editovat stávající dokumenty. V některých případech uživatel nemusí být ani přihlášen, aby mohl tyto dokumenty editovat.

### 3.2 Práce s Wiki dokumentem

Pokud chceme vytvářet nový, nebo editovat stávající wiki dokument stačí nám k tomu pouze internetový prohlížeč. Práci s editací, nebo tvořením nových dokumentů nám zpříjemní graficky přívětivé rozhraní využívající wymiwyg (What you mean is what you get) editorů a jednoduché psaní dokumentu za pomoci značkovacího jazyka(wikitextu), který zvládnou i počítačový začátečníci. Tento značkovací jazyk jak je vidět na příkladu 2.1 je mnohem jednodušší než jiné značkovací jazyky jako je například html. Vstupní text je následně přegenerován parserem do html(Hyper text markup language), který je následně zobrazen prohlížečem.

#### 3.2.1 Vytvoření wiki dokumentu

Existuje několik způsobů jak vytvořit vlastní stránku na wikipedii. Nejjednoduší z nich je zadat název dokumentu do hledání, pokud vyhledávač nenajde existující stránku, kterou bychom mohli v lepším případě pouze doplnit nabídne nám nové vytvoření, které není ničím jiným než editací prázdné stránky. Editace je popsána níže viz. kapitola 3.2.2. Ukázka vytváření nového článku na obrázku č.2



Obrázek 1: Mapa rozložení wiki serverů

### Příklad 3.1

#### Ukázka formátování textu pomocí (wikitextu)

"Lorem ipsum dolor sit amet consectetur turpis Phasellus Nam hendrerit consequat". "Quis habitasse lacus elit Ut Lorem nibh Suspendisse convallis nulla eu."

Stejný text v html:

`<i>`

Lorem ipsum dolor sit amet consectetur turpis Phasellus Nam hendrerit consequat.

`</i>`

`<strong>`

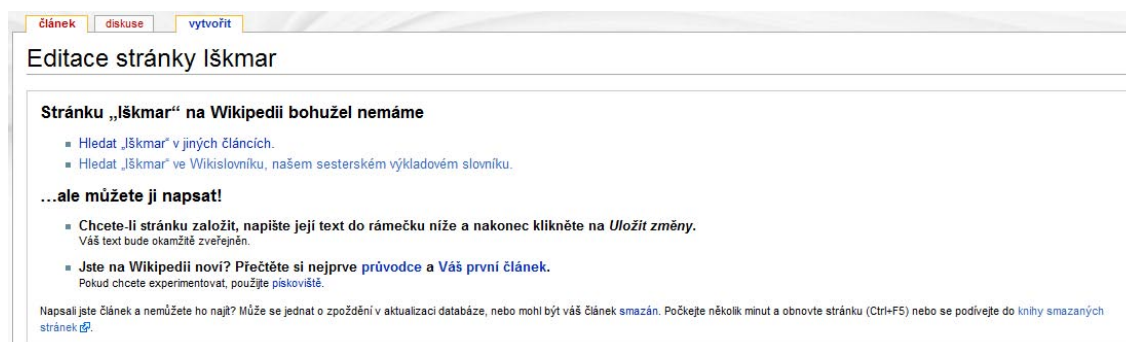
Quis habitasse lacus elit Ut Lorem nibh Suspendisse convallis nulla eu.

`</strong>`

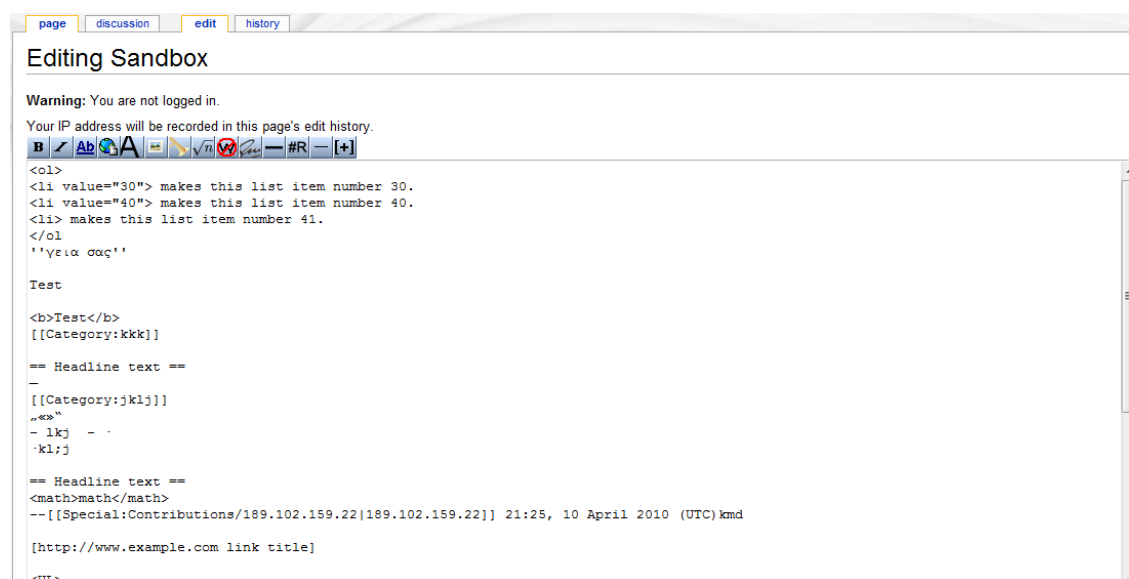
Výsledek formátování

*Lorem ipsum dolor sit amet consectetur turpis Phasellus Nam hendrerit consequat. Quis habitasse lacus elit Ut Lorem nibh Suspendisse convallis nulla eu.*

■



Obrázek 2: Vytvoření nové wiki stránky pokud nebyla nalezena ve vyhledávání



Obrázek 3: Editace wiki dokumentu

### 3.2.2 Editace wiki dokumentu

Editace dokumentu probíhá pomocí takzvaného wikipedického editoru, který nám umožní za pomoci formátovacích značek rychle a jednoduše vytvořit požadovaný výstup viz. obrázek č.3.

## 3.3 Rozšíření PermissionACL

Implementuje způsob, jak omezit přístup ke konkrétním (jmenným prostorům, stránkám, nebo kategoriím) založeným na uživatelských skupinách, nebo jménech. Tímto dosáhneme přesnějšího řízení přístupu na bázi kategorizace narozdíl od standardního řízení \$wgGroup-Permission.

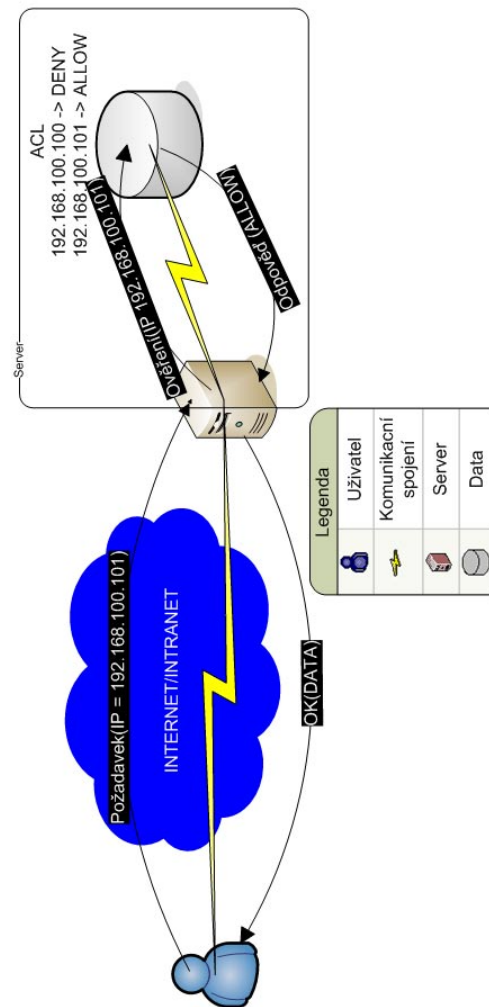
PermissionACL rozšíření je založen na ACL (Access Control List) seznamu pravidel, která se zpracovávají za sebou viz. kapitola 3.3.1. První vyhovující pravidlo se používá! Na konci seznamu je implicitní pravidlo (DENY TO ALL) zakázat všem.

### 3.3.1 Access control list

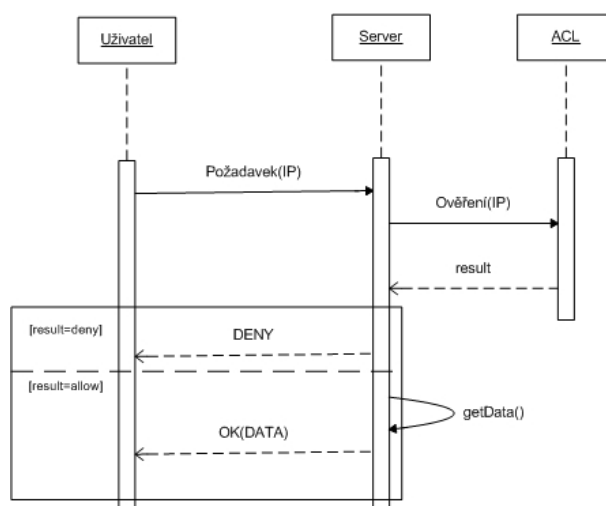
Za tímto anglickým názvem se nachází doslovně "seznam pro řízení přístupů". Obecně se jedná o mechanismus určující oprávnění uživatelů, nebo jiných objektů v závislosti na jejich nastavených právech přistupovat např. k souborům, případně databázím a možnostmi operací nad těmito soubory. V mém případě jsou to uživatelé, nebo skupiny uživatelů, kteří budou omezeni na přístup buď na jednotlivé stránky, nebo celé jmenné prostory s rozšířenými možnostmi jako jsou (editace, vytváření, mazání, přesouvání).

Pro jednodušší pochopení názorně vysvětluje funkci ACL obrázek č.4. Zde je ukázka omezení podle IP adresy uživatele, který posílá požadavek na nějaká data ze serveru. Server obdrží požadavek a prohlédne svůj ACL list obsahující IP adresy a informaci jestli této IP adrese může data poslat. V tomto případě existuje záznam pro IP 192.168.100.101 s volbou "ALLOW" tudíž server načte požadovaná data a vrátí odpověď "OK" společně s daty. Níže na obrázku č.5 je zobrazena komunikace pomocí sekvenčního diagramu, který popisuje časovou sekvenci interakce mezi objekty.

**3.3.1.1 Výběr pravidla** Výběr pravidla probíhá procházením všech pravidel dokud není nalezeno vyhovující. Pokud žádné vyhovující neexistuje je použito defaultní pravidlo, které zakáže vše.



Obrázek 4: Ukázka využití ACL



Obrázek 5: ACL Sekvenční diagram komunikace objektů



## 4 Praktická část

V této části bude popsáno vytvoření webového nástroje pro analýzu existujících a generování nových pravidel rozšíření PermissionACL v systému MediaWiki.

### 4.1 Zadání

Úkolem bylo navrhnout a realizovat webovou aplikaci, která bude umět analyzovat, generovat a editovat pravidla pro rozšíření PermissionACL v systému MediaWiki. Aplikace byla realizována pomocí dvou formulářů, jeden pro vstup a druhý pro dodatečnou editaci a vytváření nových pravidel. Uživatel si bude moci vybrat jestli chce vytvořit novou sestavu, nebo použít již existující a popřípadě ji doplnit. Při editaci pravidel bude moci uživatel měnit jejich pořadí pouhým přetáhnutím na příslušnou pozici. Podmínkou je, že na vstup budou vložena syntakticky správná pravidla, jinak je aplikace nepoužije.

### 4.2 Analýza

#### 4.2.1 Analyzování vstupního řetězce

Pro vkládání vstupních dat bylo použito formuláře. Analýza vstupních dat probíhá po zadání textového vstupu. Na vstup aplikace může být použit jakýkoliv vstup obsahující textově zapsaná pravidla. Na výstup se dostanou pouze plná a validní pravidla.

Vstupním bodem této aplikace je formulář viz. obrázek č.6.

#### 4.2.2 Uložení pravidel do mezikódu

Po odeslání formuláře pro analýzu se celý vstup uloží do pracovního souboru s názvem "soubor.php" a stránka se přesměruje na skript s PHP parserem.

Ukládání do souboru je záměrné. Důvodem je zjednodušení nasazení celé aplikace, tímto odpadá instalace databázového systému. Struktura dat vkládaných do souboru není natolik složitá a proto databázi nepotřebuje.

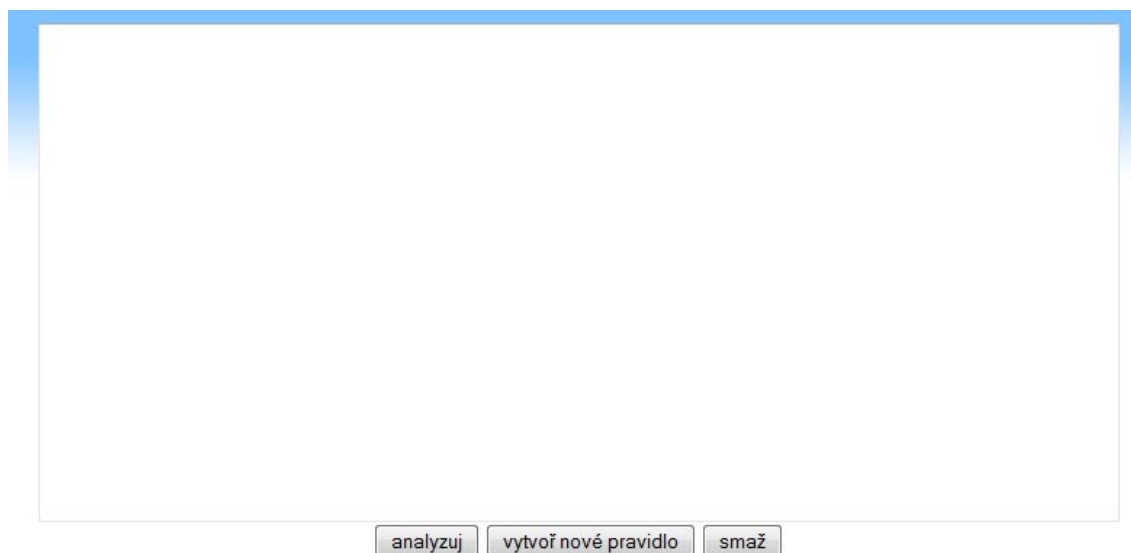
#### 4.2.3 Pear PHP Parser

Tento parser jsem volil z důvodu, že je napsaný v jazyce PHP, proto mi umožní získat výstup, který můžu dále bez větších úprav zpracovávat pomocí PHP. Tento PHP parser využívá stejnou EBNF gramatiku jako využívá PHP pro parsování sebe.

Pro zpracování pracovního souboru bylo použito statické metody "staticParseFile(string file)", které je na vstup předán parametr s cestou k pracovnímu souboru. Jako výstup je vrácen objekt obsahující pole s jednotlivými tokeny(kusy kódu) objekt má strukturu ukázanou na výpise č.2.

```
$wgPermissionACL[] = array('group' => '*',  
    'namespace' => NS_MAIN,  
    'action' => 'read',  
    'operation' => 'permit');  
  
$wgPermissionACL[] = array('group' => 'user',  
    'namespace' => array(NS_MAIN, NS_SPECIAL, NS_USER,  
        NS_CATEGORY),  
    'action' => 'read',  
    'operation' => 'permit');  
  
$wgPermissionACL[] = array('group' => 'private',  
    'namespace' => array(100, 101, 102, 103, 104, 105, 106, 107, 108, 109),  
    'action' => '*',  
    'operation' => 'permit');
```

Výpis 1: Ukázka vstupních pravidel



The image shows a web-based form for managing text rules. It consists of a large, empty rectangular text input area. Below this area, there are three buttons arranged horizontally: 'analyzuj' (analyze), 'vytvoř nové pravidlo' (create new rule), and 'smaž' (delete). The interface is simple and functional, with a light blue header bar at the top.

Obrázek 6: Formulář pro vstupní data obsahující textová pravidla

---

```
[tokens] => Array
(
    [0] => Array
    (
        [0] => 367
        [1] => Array
        (
            [0] => 370
            [1] =>
        )
    )

    [2] => Array
    (
        [0] => 309
        [1] => $wgPermissionACL
    )
)
```

---

Výpis 2: Ukázka části pole tokenů.

#### 4.2.4 Parsování vstupního textu

Na vstup parseru budou odeslána data z pracovního souboru, které parser pomocí gramatického rozkladu naplní do objektu `lex->tokens`, kde tento objekt uchovává dvourozměrné pole. Do rozměru s indexem 1 se zapíše textová hodnota aktuálního tokenu a na index 0 číselná hodnota s číselným kódem označující typ tokenu (řetězec, číslo atd.) viz ukázkový výpis č.2.

Samotná analýza probíhá procházením jednotlivých tokenů dokud není nalezen token s hodnotou `"$wgPermissionACL"` viz. výpis č.3.

---

```
for($this->i=0;$this->i<sizeof($this->tokensArray->lex->tokens);$this->i++){
    try{
        if($this->tokensArray->lex->tokens[$this->i][1]=='$wgPermissionACL'){ ...}
```

---

Výpis 3: Průchod polem tokenů.

Dál může následovat bílý znak, komentář, nebo další znak pravidla. K ověření nám slouží metoda `"check"`, která má dva vstupní parametry. První je pole číselných hodnot, které zastupují jednotlivé typy tokenů (závorky, čísla, řetězce, ...) a které musí aktuální token splnit viz. výpis č. 4.

---

```
$this->check(array(315,307,309,-1),0)
```

---

Výpis 4: Metoda ověřující správnost tokenu.

Význam jednotlivých číselných kódů je vysvětlen níže viz. tabulka č.1

Druhým parametrem se nastavuje jestli chceme pracovat s kódem tokenu, nebo s jeho hodnotou. Procházení bílými znaky nám zajistí větev viz. výpis č. 5. Pokud aktuální procházený znak je bílý, nebo se jedná o komentář, přejdeme na další znak a hledáme podle

číslený kód parseru	zastupuje
315	řetězec
-1	pole (moje vlastní značka)
307	konstanty
309	proměnné

Tabulka 1: Výpis některých kódů parseru

stejných kritérií. V případě shody se textová hodnota tokenu sloučí s textem aktuálního pravidla. Pokud metoda na daném místě nenajde přesně definovaný token, celé pravidlo se zahodí a je vyvolána vyjímka `ErrorrException` a jako parametr je jí předán aktuální token, který se zobrazí společně s hláškou "Chyba" v případě, že je nastaven debugovací režim. Tento debugovací režim lze zapnout pouze zásahem do zdrojového kódu.

```
if ($actual.token[0]==370 || $actual.token[0]==365){
    $this->i++;
    return $this->check($tokens,$option);
}
```

Výpis 5: Průchod polem tokenů.

#### 4.2.5 Uložení analyzovaných pravidel

Po analýze pravidel se vyhovující pravidla opět zapíší do již zmíněného pracovního souboru. Odtud jsou pak zobrazena uživateli v přehledném formuláři k poslední editaci.

#### 4.2.6 Mazání pravidel

Pravidla je možno mazat zaškrtnutím tlačítka s křížkem viz obr.7. Mazání se provádí ve skriptu pro uložení, proto je nutné pro smazání vyvolat funkci uložení. Na stránku se skriptem je odesláno pole "DEL", které má jako klíčové položky identifikační čísla mazaných pravidel. Pomocí rozdílu klíčových hodnot dvou polí eliminujeme "id", která se nemají vložit a provedeme uložení.

#### 4.2.7 Editace pravidel

Editace probíhá pomocí formuláře viz. obrázek č. 7. Editovat můžeme nejen obsah pravidel, ale také jejich pořadí, které provedeme přetáhnutím jednotlivých pravidel. Tohoto je docíleno využitím javascriptového frameworku, který nám umožní velice jednoduše pracovat s DOM objekty.

Obrázek 7: Obrázek s formulářem pro editaci pravidel

```

$wgPermissionACL[0] = array(
  'user' => array('dur30','rad047'),
  'namespace' => array(122,123),
  'action' => '*',
  'operation' => 'permit'
);

$wgPermissionACL[1] = array(
  'user' => array('dur30','gau003'),
  'namespace' => array(124,125),
  'action' => '*',
  'operation' => 'permit'
);

```

Obrázek 8: Export pravidel v textové podobě

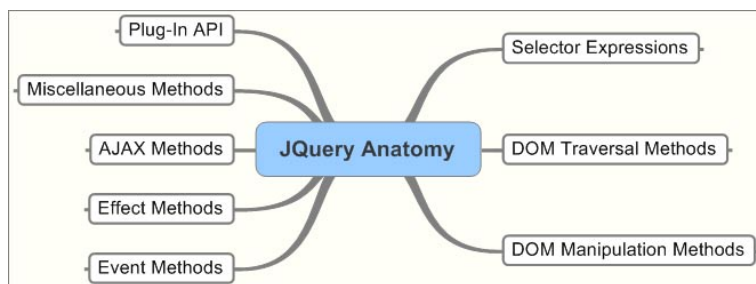
#### 4.2.8 Export editovaných pravidel

Export pravidel je důležitou součástí aplikace pro finální nasazení pravidel na server. Dle požadavků se pravidla exportují textově do přehledné podoby viz obr. 8.

#### 4.2.9 Javascript JQuery Framework

Je nástroj (knihovna), který usnadňuje práci s kódem. Jelikož se jedná o javascriptový framework, nástroj slouží pro práci s javascriptem. Tento nástroj již obsahuje řadu hotových řešení, které se dají dále použít. V mém projektu jsem použil plugin nazývaný se "sortable", který mi umožňuje měnit pořadí jednotlivých pravidel. Tento plugin ale umožňuje pouze vizuální přehození pravidel, neřeší však jejich další uložení do databáze. Ukázka č.7 inicializace sortable.





Obrázek 9: Architektura JQuery

### 4.3 Uložení editovaných pravidel

Ukládání pravidel je řešeno pomocí vlastnosti serializace pluginu sortable. Serializace převede veškerá id html elementů do podoby viz. výpis č.8. Touto metodou získáme aktuální pořadí pravidel, které při zapisování do souboru zapíšeme každému pravidlu jako index, tímto docílíme uložení změn pořadí.

---

```
<li id='pravidla-1'>
    ... zde obsah jednoho pravidla ...
</li>
<li id='pravidla-2'>
    ... zde obsah jednoho pravidla ...
</li>
```

bude převedeno na "pravidla[]=1\&pravidla[]=2".

---

#### Výpis 8: Ukázka serializace pravidel

JQUERY disponuje funkcemi jako :

- Výběr DOM elementů pomocí otevřeného cross-browser selektorového enginu Sizzle, odnože projektu jQuery
- Funkce pro procházení a změnu DOM (včetně podpory pro 1–3 a základní XPath)
- Události
- Manipulace s CSS
- Efekty a animace
- AJAX
- Rozšiřitelnost
- Utility – např. informace o prohlížeči nebo funkce each
- Javascriptové pluginy

## 4.4 Grafické zpracování

Pro úpravu grafického vzhledu je použito kaskádových stylů. Cascading Style Sheets (CSS) je jednoduchý mechanismus pro přidávání stylů (např. fonty, barvy, odsazování) webovým dokumentům. S CSS styly můžeme pomocí jednoho souboru ovlivňovat design celého webu. Avšak kaskádové styly lze vkládat také in-line přímo do souboru s HTML stránkou. V této práci bylo použito pár základních stylů, existují však možnosti propojení CSS s javascriptem k docílení dynamického stylování, kterému se říká DHTML (Dynamic Hyper Text Markup Language). Výpis CSS stylů z bakalářské práce viz. 9



---

```
<style type="text/css">
* {
  margin:0;
  padding:0;
}
body {
  font-family: Arial , Helvetica, sans-serif;
  font-size: 16px;
  margin-top: 10px;
}
#content {
  width: 800px;
  margin: auto;
  height: auto;
  overflow: hidden;
}
#content ul{
  text-align: left ;
}
#content li {
  list-style: none;
  margin: 0 0 4px 0;
  padding: 10px;
  background-color:#00CCCC;
  border: #CCCCCC solid 1px;
  color:# fff ;
}
</style>
```

---

Výpis 9: Ukázka css stylů

## 4.5 Vývojové prostředí

Všechn kód byl programován v prostředí Eclipse IDE s doinstalovanou podporou pro PHP. Velká výhoda tohoto prostředí je lazení skriptů za pomoci krokování, které umožňuje například sledovat hodnoty proměnných. Další velkou výhodou je automatické barvení syntaxe kódu, zabalování bloků kódu na jeden řádek, našeptávání názvů metod a proměnných.

Grafický výstup byl převážně laděn v prohlížeči Mozilla Firefox za použití rozšiřujícího pluginu FireBug, který za pomoci DOM(Document object model) umí po najetí na libovolný element v dokumentu zobrazit jeho styl a také celý DOM strom.

## 5 Závěr

V bakalářské práci je shrnuta práce s dokumenty Mediawiki, to znamená jejich editace a vytváření. Dále je zde popsána multiplatformní aplikace analyzující textový vstup, který je převeden na grafické uživatelské rozhraní.

Výsledkem bakalářské práce je aplikace pro tvorbu a editaci permissions ACL pravidel. Aplikace je rozdělena do dvou částí. První část slouží pro zadání vstupu k analýze. Ve druhé části je možno tento vstup dále editovat, nebo vytvořit kompletně nová pravidla v graficky přívětivém uživatelském prostředí.

Pro implementaci bylo použito javascriptového frameworku JQuery a skriptovacího jazyka PHP.

Do systému je možno doimplementovat podporu databází, která by zajistila jednodušší práci s ukládáním a načítáním dat. Přibyl by však problém s doinstalací databázového systému.

## 6 Reference

- [1] Internet, *http://www.mediawiki.cz*
- [2] Internet, *http://www.jquery.com*
- [3] Internet, *http://www.kosek.cz*
- [4] Internet, *http://www.jakpsatweb.cz*
- [5] Internet, *http://www.php.net*
- [6] Internet, *http://pear.php.net*

## A Obsah přiloženého CD

Adresář	Obsah
/Zdroj/	Zdrojové kódy
/Přílohy/	Adresář se soubory příloh